PROBLEM SOLVING AND TESTING USING JAVA

WEEK 1 – TASKS

NAME : K .Kotesh

VTU NO : VTU25857

Task:1

Iseven

Aim:- To write a Java program that checks whether a given integer is even or odd.

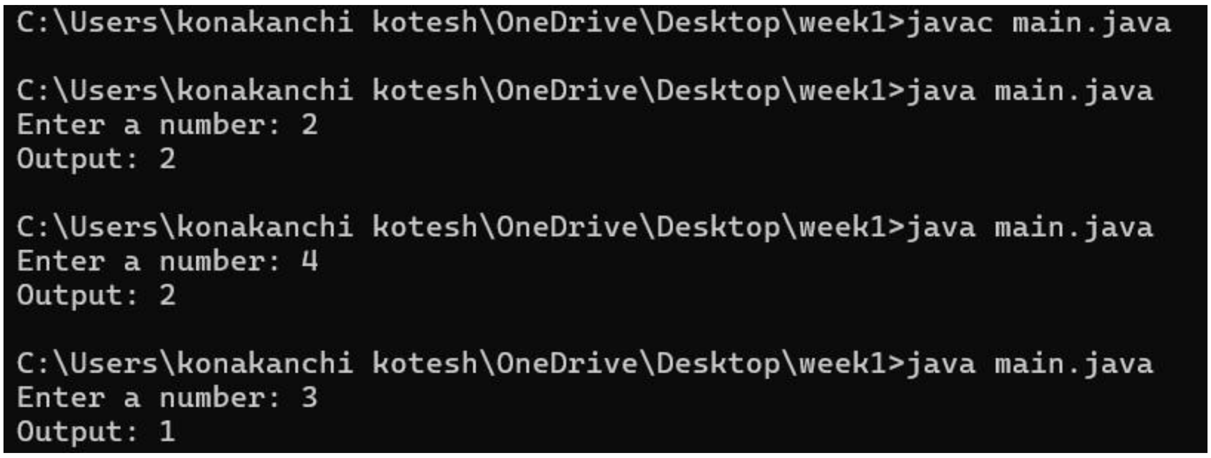
Algorithm:-

1. Start.
2. Take an integer input (input1).
3. Compute input 1 % 2.
4. If remainder = 0 → number is even → return 2.
5. Else → number is odd → return 1.
6. End.

Program:-

```
import java.util.Scanner;


public class main {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);
System.out.print("Enter a number: ");         int
input1 = sc.nextInt();


        UserMainCode obj = new UserMainCode();
        int result = obj.isEven(input1);


        System.out.println("Output: " + result);
    }
}
class UserMainCode {
    public int isEven(int input1) {
if (input1 % 2 == 0) {           return
2;         } else {             return 1;
        }
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac main.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java main.java
Enter a number: 2
Output: 2

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java main.java
Enter a number: 4
Output: 2

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java main.java
Enter a number: 3
Output: 1
```

Result:- Thus, To write a Java program that checks whether a given integer is even or odd is are successfully executed.

Task2:

Access and print the element at a given index in an array.

Aim:- To write a Java program that takes an array and an index as input, then prints the element present at that index.

Algorithm:-

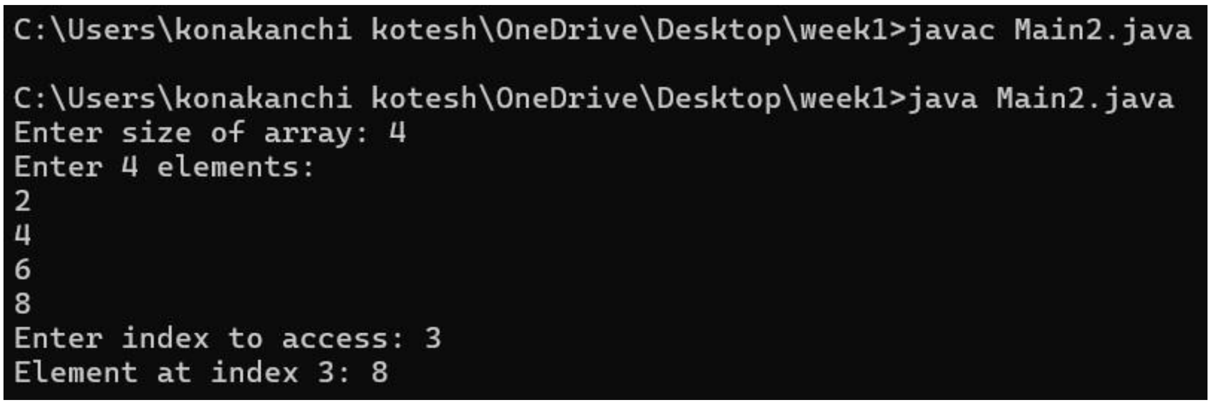
1. Start.
2. Read the size of the array.
3. Read the array elements.
4. Read the index to be accessed.
5. Check if the index is valid (0 <= index < array.length).
6. If valid → print the element at that index.
7. Else → print an error message.
8. End.

Program:-

```
import java.util.Scanner;


public class main2 {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter size of array: ");
int n = sc.nextInt();


        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
```

```
        }

        System.out.print("Enter index to access: ");
int index = sc.nextInt();

        if (index >= 0 && index < n) {
            System.out.println("Element at index " + index + ": " + arr[index]);
} else {
            System.out.println("Invalid index!");
        }
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main2.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main2.java
Enter size of array: 4
Enter 4 elements:
2
4
6
8
Enter index to access: 3
Element at index 3: 8
```

Result:- Thus, To write a Java program that takes an array and an index as input, then prints the element present at that index is are executed successfully.

Task3:

Search for a given element in a sorted array using Binary Search

Aim:- To write a Java program that searches for a given element in a sorted array using the Binary Search algorithm.

Algorithm:-

1. Start
2. Read the size of the array and its elements (sorted order).
3. Read the element to be searched (key).
4. Initialize low = 0, high = n-1.
5. While low <= high:
6. Compute mid = (low + high) / 2.
7. If arr[mid] == key → return mid.
8. If arr[mid] < key → search in right half (low = mid + 1).
9. Else → search in left half (high = mid - 1).
10. If not found → return -1.
11. End

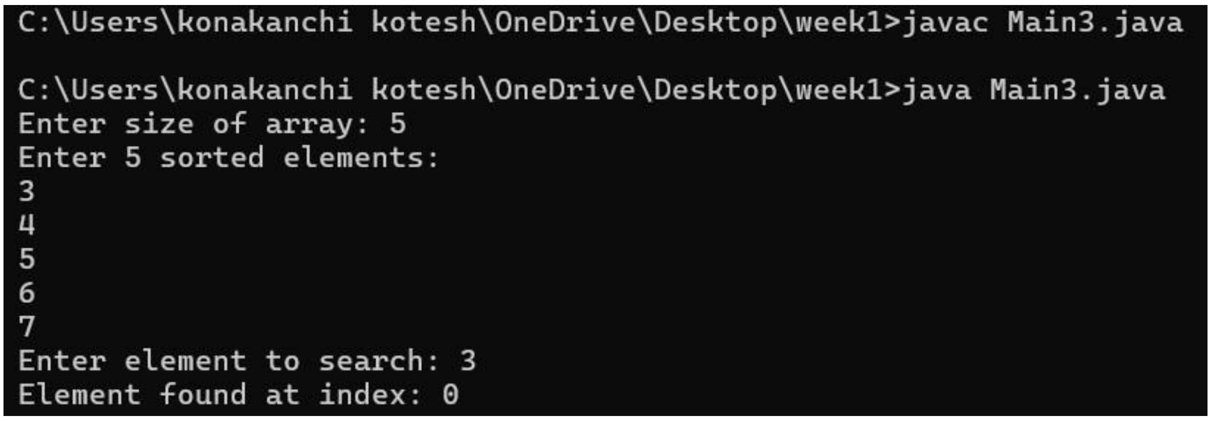
Program:-

```
import java.util.Scanner;


public class Main3 {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter size of array: ");
int n = sc.nextInt();
```

```
        int[] arr = new int[n];
        System.out.println("Enter " + n + " sorted elements:");
        for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
        }

        System.out.print("Enter element to search: ");
int key = sc.nextInt();

        int result = binarySearch(arr, key);

        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found!");
        }
    }

    public static int binarySearch(int[] arr, int key) {
int low = 0, high = arr.length - 1;

        while (low <= high) {
int mid = (low + high) / 2;

            if (arr[mid] == key) {
return mid;
```

```
            } else if (arr[mid] < key) {
low = mid + 1;
            } else {
high = mid - 1;
            }
}
        return -1;
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main3.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main3.java
Enter size of array: 5
Enter 5 sorted elements:
3
4
5
6
7
Enter element to search: 3
Element found at index: 0
```

Result:- Thus, To write a Java program that searches for a given element in a sorted array using the Binary Search algorithm is are executed successfully.

Task4:

Find the maximum element in an array of n integers

Aim:- To write a Java program that finds and prints the maximum element in an array of n integers.

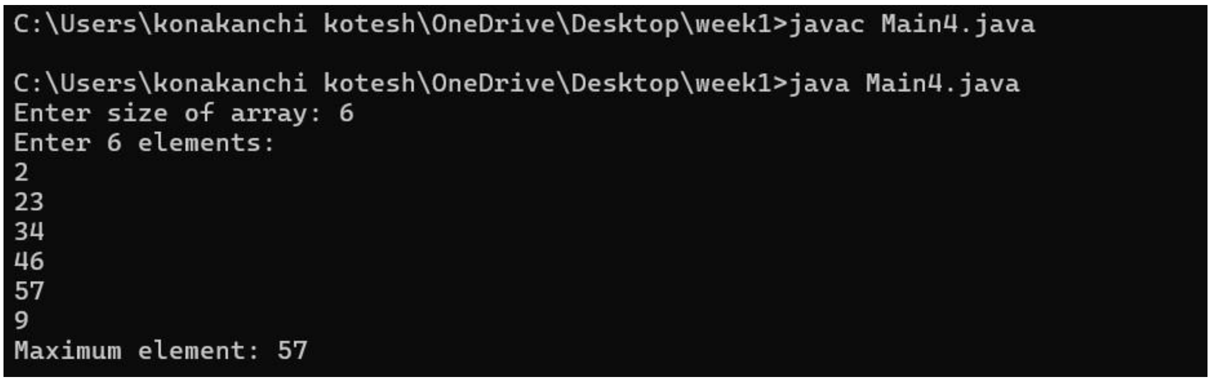
Algorithm:-

1. Start.
2. Read the size of the array (n).
3. Read n integers into the array.
4. Initialize max = arr[0].
5. Traverse the array from index 1 to n-1:
6. If arr[i] > max → update max = arr[i].
7. Print max.
8. End.

Program:-

```
import java.util.Scanner;


public class Main {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter size of array: ");
int n = sc.nextInt();


        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int max = arr[0];         for
(int i = 1; i < n; i++) {
if (arr[i] > max) {
max = arr[i];
            }
        }

        System.out.println("Maximum element: " + max);
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main4.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main4.java
Enter size of array: 6
Enter 6 elements:
2
23
34
46
57
9
Maximum element: 57
```

Result:- Thus, To write a Java program that finds and prints the maximum element in an array of n integers is are executed successfully.

Task5:

Kth smallest element

Aim:- To write a Java program that finds the Kth smallest element in an array of integers.

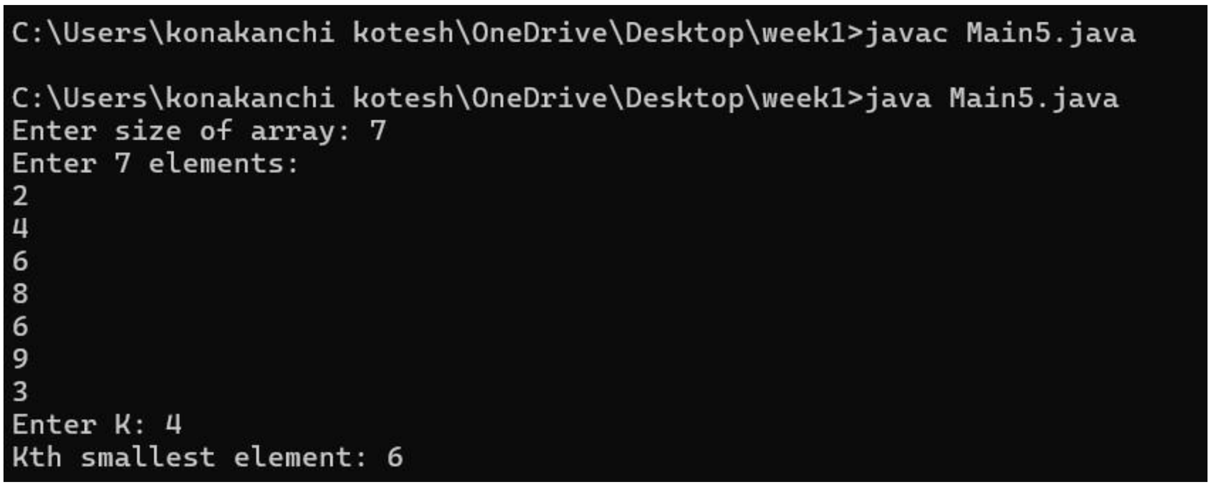
Algorithm:-

1. Start.
2. Read the size of the array (n).
3. Read n integers into the array.
4. Read the value of K.
5. Sort the array in ascending order.
6. Access the element at index K-1 (since arrays are 0-based).
7. Print that element.
8. End.

Program:-

```
import java.util.Scanner; import
java.util.Arrays;
public class Main5 {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter size of array: ");
int n = sc.nextInt();


        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
        }

        System.out.print("Enter K: ");
int k = sc.nextInt();

        Arrays.sort(arr);

        if (k > 0 && k <= n) {
            System.out.println("Kth smallest element: " + arr[k - 1]);
        } else {
            System.out.println("Invalid K!");
        }
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main5.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main5.java
Enter size of array: 7
Enter 7 elements:
2
4
6
8
6
9
3
Enter K: 4
Kth smallest element: 6
```

Result:- Thus, To write a Java program that finds the Kth smallest element in an array of integers is are executed successfully.

Task6:

Print all possible pairs of elements from an array of size n

Aim:- To write a Java program that prints all possible pairs of elements from an array of size n.

Algorithm:-

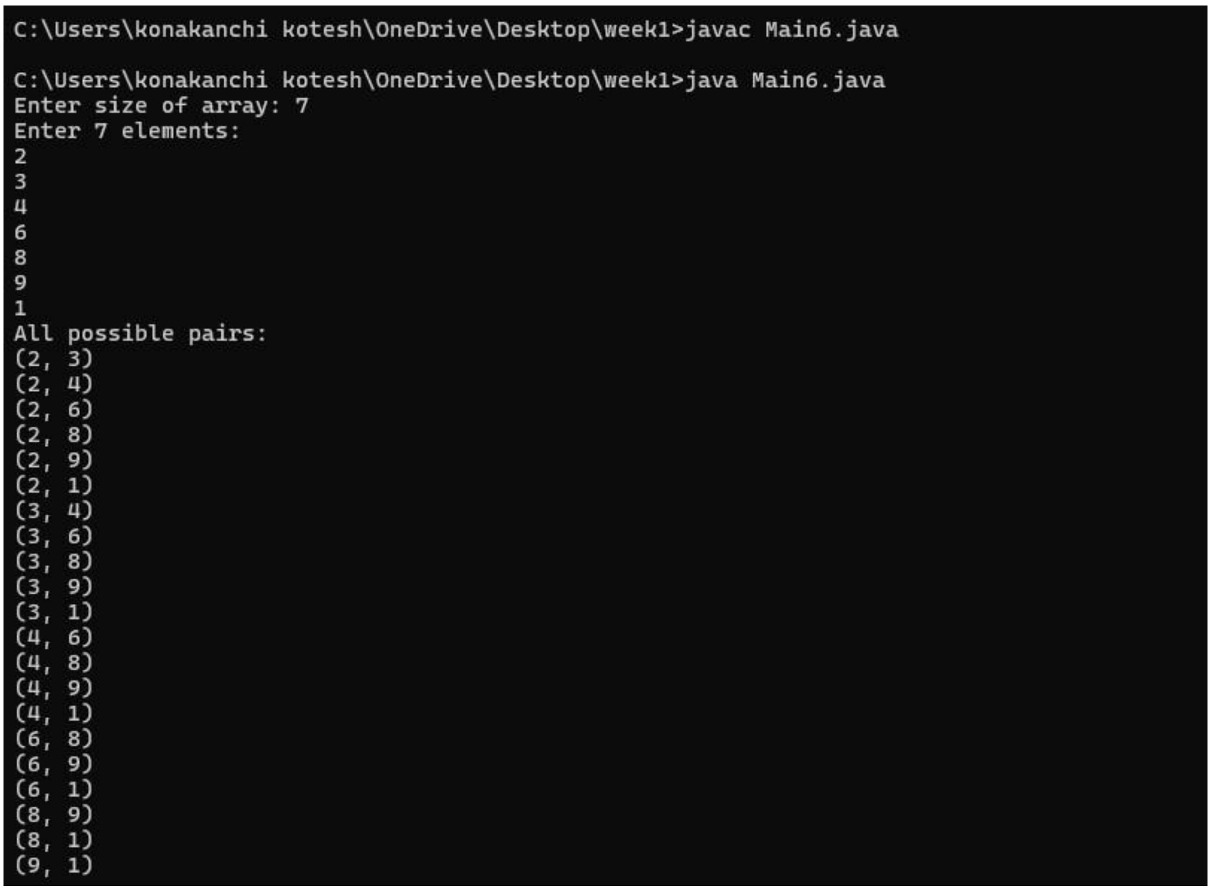
1. Start.
2. Read the size of the array (n).
3. Read n integers into the array.
4. Use two nested loops:
5. Outer loop runs from i = 0 to n-1.
6. Inner loop runs from j = i+1 to n-1.
7. Print the pair (arr[i], arr[j]).
8. End.

Program:-

```
import java.util.Scanner;


public class Main6 {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter size of array: ");
int n = sc.nextInt();


        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
```

```
        }


        System.out.println("All possible pairs:");
for (int i = 0; i < n; i++) {            for (int j = i
+ 1; j < n; j++) {
                System.out.println("(" + arr[i] + ", " + arr[j] + ")");
            }
        }
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main6.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main6.java
Enter size of array: 7
Enter 7 elements:
2
3
4
6
8
9
1
All possible pairs:
(2, 3)
(2, 4)
(2, 6)
(2, 8)
(2, 9)
(2, 1)
(3, 4)
(3, 6)
(3, 8)
(3, 9)
(3, 1)
(4, 6)
(4, 8)
(4, 9)
(4, 1)
(6, 8)
(6, 9)
(6, 1)
(8, 9)
(8, 1)
(9, 1)
```

Result:- Thus,To write a Java program that prints all possible pairs of elements from an array of size n is are successfully executed.

Task7:

DigitSum opt: sum of even or odd digits

Aim:- To write a Java program that calculates the sum of digits of a given number based on an option:1 or 2.

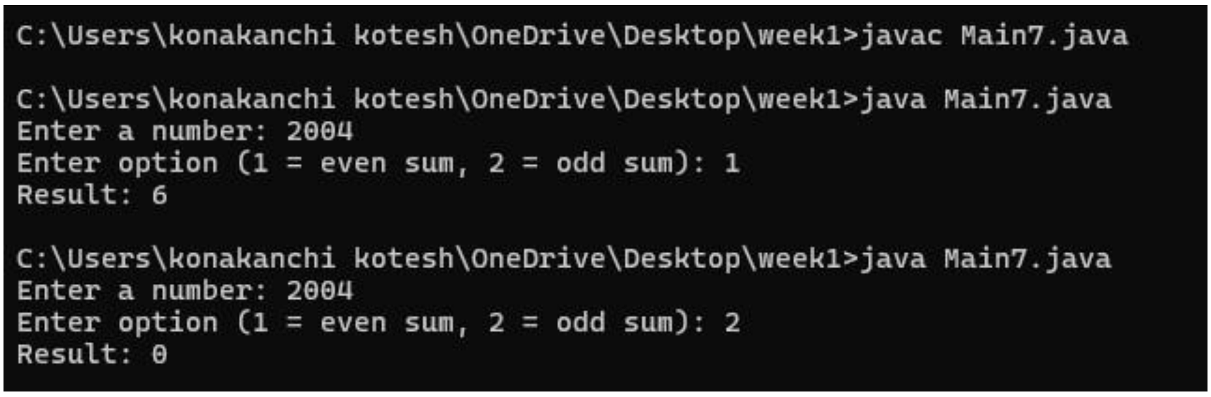
Algorithm:-

1. Start.
2. Read an integer number (num).
3. Read the option (opt).
4. If opt = 1 → sum even digits.
5. If opt = 2 → sum odd digits.
6. Extract digits one by one using % 10 and / 10.
7. Check each digit:
8. If digit is even and opt = 1 → add to sum.
9. If digit is odd and opt = 2 → add to sum.
10. Print the sum.
11. End.

Program:-

```
import java.util.Scanner; public class
Main {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);
System.out.print("Enter a number: ");         int
num = sc.nextInt();
        System.out.print("Enter option (1 = even sum, 2 = odd sum): ");
        int opt = sc.nextInt();
        int sum = digitSum(num, opt);
        System.out.println("Result: " + sum);
    }
    public static int digitSum(int num, int opt) {
int sum = 0;        while (num > 0) {            int
digit = num % 10;            if (opt == 1 &&
digit % 2 == 0) {
                sum += digit;
            } else if (opt == 2 && digit % 2 != 0) {
sum += digit;
            }
            num /= 10;
        }
        return sum;
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main7.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main7.java
Enter a number: 2004
Enter option (1 = even sum, 2 = odd sum): 1
Result: 6

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main7.java
Enter a number: 2004
Enter option (1 = even sum, 2 = odd sum): 2
Result: 0
```

Result:- Thus,To write a Java program that calculates the sum of digits of a given number based on an option: 1 or 2 is are successfully executed.

Task8:

Nth Fibonacci

Aim:- To write a Java program that computes the Nth Fibonacci number. Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, … Formula: F(n) = F(n-1) + F(n-2).

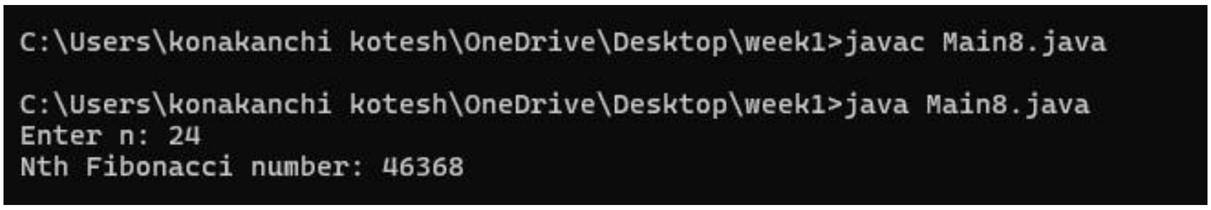
Algorithm:-

1. Start.
2. Read the integer n.
3. If n == 0 → return 0.
4. If n == 1 → return 1.
5. Otherwise, use iteration:
6. Initialize a = 0, b = 1.
7. Loop from 2 to n:
8. c = a + b
9. Update a = b, b = c.
10. Result = b.
11. Print the result.
12. End. **Program:-**

```
import java.util.Scanner; public class
Main {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);
System.out.print("Enter n: ");         int n =
sc.nextInt();
        int result = nthFibonacci(n);
        System.out.println("Nth Fibonacci number: " + result);
    }
```

```
   public static int nthFibonacci(int n) {
if (n == 0) return 0;         if (n == 1)
return 1;         int a = 0, b = 1, c = 0;
for (int i = 2; i <= n; i++) {             c =
a + b;             a = b;             b = c;
        }
        return b;
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main8.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main8.java
Enter n: 24
Nth Fibonacci number: 46368
```

Result:- Thus, To write a Java program that computes the Nth Fibonacci number is are successfully executed.

Task9:

Palindrome Number

Aim:- To write a Java program that checks whether a given integer is a palindrome number.

Algorithm:-

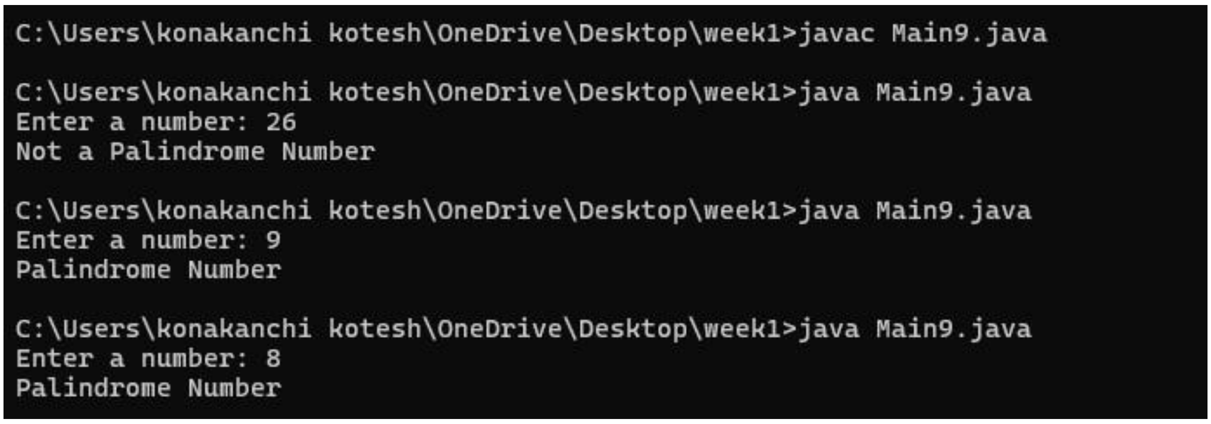
1. Start.
2. Read an integer num.
3. Store the original number in temp.
4. Reverse the digits of num:
5. Initialize rev = 0.
6. While num > 0:
7. Extract digit → digit = num % 10.
8. Update reverse → rev = rev * 10 + digit.
9. Reduce number → num = num / 10.
10. Compare rev with temp.
11. If equal → number is palindrome.
12. Else → not palindrome.
13. End.

Program:-

```
import java.util.Scanner; public class
Main9 {    public static void
main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter a number: "); int
      num = sc.nextInt();
```

```
        if (isPalindrome(num)) {
            System.out.println("Palindrome Number");
        } else {
            System.out.println("Not a Palindrome Number");
        }
    }
    public static boolean isPalindrome(int num) {
int temp = num;
        int rev = 0;
        while (num > 0) {
int digit = num % 10;
rev = rev * 10 + digit;
num /= 10;
        }
        return temp == rev;
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main9.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main9.java
Enter a number: 26
Not a Palindrome Number

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main9.java
Enter a number: 9
Palindrome Number

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main9.java
Enter a number: 8
Palindrome Number
```

Result:-Thus,To write a Java program that checks whether a given integer is a palindrome number is are successfully executed.

Task10:

Sum of last digit of two given numbers

Aim:- To write a Java program that reads two integers and prints the sum of their last digits.

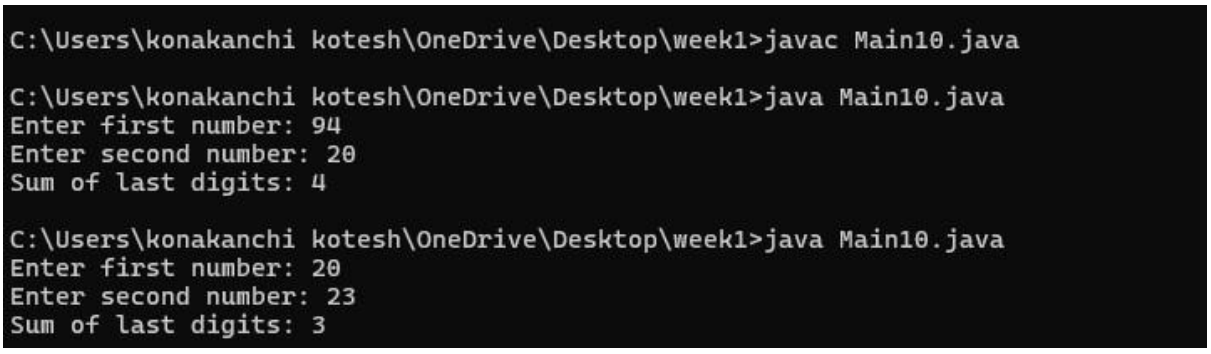
Algorithm:-

1. Start.
2. Read two integers (num1, num2).
3. Extract the last digit of each number using % 10.
3. last1 = num1 % 10
4. last2 = num2 % 10 5. Compute sum = last1 + last2.
6. Print the result.
7. End.

Program:-

```
import java.util.Scanner; public class
Main10 {     public static void
main(String[] args) {
        Scanner sc = new Scanner(System.in);
System.out.print("Enter first number: ");         int
num1 = sc.nextInt();
        System.out.print("Enter second number: ");
int num2 = sc.nextInt();         int last1 = num1 %
10;         int last2 = num2 % 10;
        int sum = last1 + last2;
        System.out.println("Sum of last digits: " + sum);
    }
}
```

Output:-



```
C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>javac Main10.java

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main10.java
Enter first number: 94
Enter second number: 20
Sum of last digits: 4

C:\Users\konakanchi kotesh\OneDrive\Desktop\week1>java Main10.java
Enter first number: 20
Enter second number: 23
Sum of last digits: 3
```

Result:- Thus,To write a Java program that reads two integers and prints the sum of their last digits is are successfully executed.